PRIVACY PROTECTION ON RFID DATA PUBLISHING

MING CAO

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS

SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

AUGUST 2009

Your file Votre référence
ISBN: 978-0-494-63109-6
Our file Notre référence
ISBN: 978-0-494-63109-6

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Ming Cao**

Entitled: **Privacy Protection on RFID Data Publishing**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science in Information Systems Security

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

_____ Examiner

_____ Examiner

_____ Examiner

_____ Supervisor

_____ Co-supervisor

Approved _____

Chair of Department or Graduate Program Director

_____ 20 _____ _____

Dr. Robin A.L. Drew, Dean

Faculty of Engineering and Computer Science

Abstract

Privacy Protection on RFID Data Publishing

Ming Cao

Radio Frequency IDentification (RFID) is a technology of automatic object identification. Retailers and manufacturers have created compelling business cases for deploying RFID in their supply chains. Yet, the uniquely identifiable objects pose a privacy threat to individuals. In this paper, we study the privacy threats caused by publishing RFID data. Even if the explicit identifying information, such as name and social security number, has been removed from the published RFID data, an adversary may identify a target victim's record or infer her sensitive value by matching *a priori* known visited locations and time. RFID data by its nature is high-dimensional and sparse, so applying traditional k-anonymity to RFID data suffers from the curse of high-dimensionality, and results in poor information usefulness. We define a new privacy model and develop an anonymization algorithm to accommodate special challenges on RFID data. Then, we evaluate its effectiveness on synthetic data sets.

Acknowledgments

I would like to express my sincerest gratitude to my supervisor, Dr. Benjamin Fung, who has supported me throughout my studies with his patience and knowledge. Most importantly, he advised me with unflinching encouragement and support during the researcher process. My master's degree would not have been completed without his guidance, help and constructive criticism.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 RFID Data Publishing

Radio Frequency IDentification (RFID) is a technology for automatic identification of single or bulk objects from a distance, using radio signals. RFID was first introduced during World War II for distinguishing enemy planes from allied planes. Until recently the cost of building a RFID infrastructure was viewed as being too expensive for commercial and civil applications. RFID has wide applications in many areas including manufacturing, healthcare, and transportation. Figure 1 depicts an overview of a RFID information system, typically consisting of a large number of tags and readers and an infrastructure for handling high volumes of RFID data. As depicted in the figure, a tag is a small device that can be attached to an object, such as a person or a manufactured item, for the purpose of unique identification. A reader is an electronic device positioned in a strategic location, such as a warehouse loading bay or a subway station entrance, that communicates with the

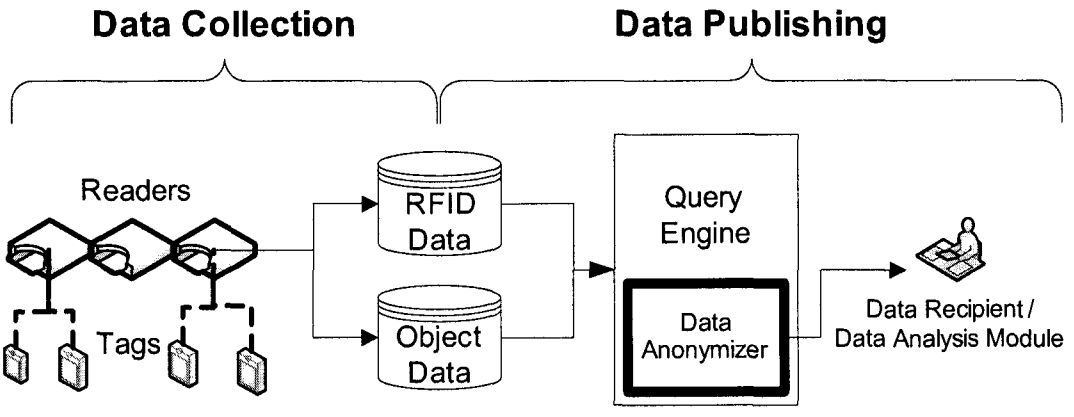


Figure 1: Data Flow in RFID System

RFID tag. A reader broadcasts a radio signal to the tag, which then transmits its information back to the reader [47]. Streams of RFID data records, in the format of $\langle EPC, loc, t \rangle$, are then stored in a RFID database, where EPC (Electronic Product Code) is a unique identifier of the tagged object, loc is the location of the reader, and t is the time of detection. A data recipient (or a data analysis module) can obtain the information on either specific tagged objects or general workflow patterns [24] by submitting data requests to the query engine. The query engine then responds to the requests by joining the RFID data with some object-specific data.

Retailers and manufacturers have created compelling business cases for deploying RFID in their supply chains, from reducing out-of-stocks at Wal-Mart to up-selling consumers in Prada. Yet, the uniquely identifiable objects pose a privacy threat to individuals, such as tracing a person's movements, and profiling individuals become possible. Most previous work on privacy-preserving RFID technology [47] focused on the threats caused by the physical RFID tags. They proposed techniques like EPC re-encryption and killing tags [30]

to address the privacy issues in the *data collection* phase, but these techniques cannot address the privacy threat in the *data publishing* phase, when a large volume of RFID data is released to a third party.

In this thesis, we study the privacy threats in the data publishing phase and define a practical privacy model to accommodate the special challenges of RFID data. We propose an anonymization algorithm (the data anonymizer in Figure 1) to transform the underlying raw object-specific RFID data into a version that is immunized against privacy attacks. The term "publishing" has a broad sense here. It includes sharing the RFID data with specific recipients and releasing data for public download. The general assumption is that the recipient could be an attacker, who attempts to associate a target victim (or multiple victims) to some sensitive information from the published data.

1.2 Privacy Threats in RFID Systems

There are many real-life examples of RFID data publishing in healthcare [57]. Recently, some hospitals have adopted RFID sensor systems to track the positions of their patients, doctors, medical equipments, and devices inside a hospital, with the goals of minimizing medical errors and improving the management of patients and resources. Analyzing RFID data, however, is a non-trivial task. The hospital management often does not have the expertise to perform the analysis themselves but outsource this process and, therefore, requires granting a third party access to the RFID and patient data. The following example illustrates the privacy threats caused by publishing RFID data.

Table 1: Raw patient-specific path table T

EPC	Path	Diagnosis	...
1	$\langle a1 \rightarrow d2 \rightarrow b3 \rightarrow e4 \rightarrow f6 \rightarrow c7 \rangle$	HIV	
2	$\langle b3 \rightarrow e4 \rightarrow f6 \rightarrow e8 \rangle$	Flu	
3	$\langle b3 \rightarrow c7 \rightarrow e8 \rangle$	Flu	
4	$\langle d2 \rightarrow f6 \rightarrow c7 \rightarrow e8 \rangle$	Allergy	
5	$\langle d2 \rightarrow c5 \rightarrow f6 \rightarrow c7 \rangle$	HIV	
6	$\langle c5 \rightarrow f6 \rightarrow e9 \rangle$	Allergy	
7	$\langle d2 \rightarrow c5 \rightarrow c7 \rightarrow e9 \rangle$	Fever	
8	$\langle f6 \rightarrow c7 \rightarrow e9 \rangle$	Fever	

Example 1.2.1. A hospital wants to release the patient-specific path table, Table 1, to a third party for data analysis. Explicit identifiers, such as patient names and EPC, are removed. Each record contains a *path* and some patient-specific information, where a *path* contains a sequence of *pairs* $(loc_i t_i)$ indicating the patient's visited location loc_i at timestamp t_i. For example, $EPC\#3$ has a path $\langle b3 \rightarrow c7 \rightarrow e8 \rangle$, meaning that the patient has visited locations b, c, and e at timestamps 3, 7, and 8, respectively. Without loss of generality, we assume that each data record contains only one sensitive attribute, namely diagnosis, in this example.

One data recipient, who is an attacker, seeks to identify the record and/or sensitive value of a target victim from the published data. We focus on two types of privacy attacks:

1. *Record linking*: if a path in the table is so specific that not many people match it, releasing the RFID data may lead to linking the victim's record, and therefore, her contracted diagnosis. Suppose that the attacker knows that the target victim, Alice, has visited e and c at timestamps 4 and 7, respectively. Alice's record, together with her sensitive value (HIV in this case), can be uniquely identified because $EPC\#1$ is the *only* record that contains $e4$ and $c7$.

2. *Attribute linking*: if a sensitive value occurs frequently together with some combination of pairs, then the sensitive information can be inferred from such combination even though the exact record of the victim cannot be identified. Suppose the attacker knows that another target victim, Bob, has visited $d2$ and $f6$. Since two out of the three records ($EPC\#1,4,5$) containing $d2$ and $f6$ have sensitive value HIV, the attacker can infer that Bob has HIV with $2/3 = 67\%$ confidence.

Many privacy models, such as K-anonymity [7] [19] [20] [33] [45] [53], ℓ-diversity [38], confidence bounding [55] [56], and t-closeness [36] have been proposed to thwart privacy threats caused by record linking and attribute linking in the context of relational databases. All these works assume a given set of attributes called *quasi-identifier* (QID) that can identify an individual. Although these privacy models are effective for anonymization on relational databases, they are not applicable to RFID data due to two special challenges posed by RFID data:

High-Dimensionality: RFID data by default is high-dimensional due to the large combinations of locations and timestamps. Consider a hospital having 50 rooms that operate 12 hours per day. The RFID data table would have $50 \times 12 = 600$ dimensions. Each dimension could be a potential quasi-identifying (QID) attribute used for record or attribute linking. Traditional privacy model, say K-anonymity, would include all dimensions into a single QID and require every path to be shared by at least K records. Due to the curse of high-dimensionality [3], it is very likely that a lot of data has to be suppressed in order to satisfy K-anonymity. For example, to achieve 2-anonymity in Table 1, $a1, d2, b3, e4, c7, e9$ have to be suppressed even if K is small. Such anonymous data becomes useless for data

Table 2: Anonymous table T' for L=2, K=2, C=50%

EPC	Path	Diagnosis	...
1	$\langle b3 \rightarrow f6 \rightarrow c7 \rangle$	HIV	
2	$\langle b3 \rightarrow f6 \rightarrow e8 \rangle$	Flu	
3	$\langle b3 \rightarrow c7 \rightarrow e8 \rangle$	Flu	
4	$\langle f6 \rightarrow c7 \rightarrow e8 \rangle$	Allergy	
5	$\langle c5 \rightarrow f6 \rightarrow c7 \rangle$	HIV	
6	$\langle c5 \rightarrow f6 \rightarrow e9 \rangle$	Allergy	
7	$\langle c5 \rightarrow c7 \rightarrow e9 \rangle$	Fever	
8	$\langle f6 \rightarrow c7 \rightarrow e9 \rangle$	Fever	

analysis.

Data Sparseness: RFID data is usually sparse. Consider patients in a hospital or patients in a public transit system. They usually visit only few locations compared to all available locations, so each RFID path is relatively short. Anonymizing these short paths in a high-dimensional space poses great challenge for traditional anonymization techniques because the paths have little overlap. Enforcing K-anonymity on sparse data would render the data useless.

A recent work [61] on transaction data anonymization also utilized a similar assumption that attacker know at most h items to bound the prior knowledge of an adversary, but their privacy models together with their methods are not applicable to anonymizing path data of moving objects.

1.3 Objective and Motivation

We motivate the problem with a real-life example of sharing person-specific RFID data. The Oyster Travelcard Transport for London (TfL), is a successful application of RFID technology in transit system. Passengers register their personal information when they first

purchase their RFID-tagged smart cards. Then, the appropriate fare amount is deducted from their cards every time they use the transport services. Passengers refill their smart card anytime as needed. The public transit companies utilize the personal journey data (the RFID data) to improve their services. Analyzing RFID data is a non-trivial task; transit companies often do not have the expertise to perform the analysis themselves but outsource this process and, therefore, require granting a third party access to the RFID data and passenger data (object data in Figure 1). The passengersŠ data may contain person-specific (sensitive) information, such as age, disability status, and (un)employment status. TfL does say that it does not associate journey data with named passengers, although they provide such data to government agencies on request [52]. Our goal is to answer the question: how can RFID data holders (e.g., the transit company) safeguard data privacy while keeping the released RFID data useful?

Traditional K-anonymity and its extended privacy models assume that a QID contains all attributes (dimensions) because the attacker could potentially use any or even all QID attributes as prior knowledge to perform record or attribute linking. However, in real-life privacy attacks, it is unlikely that an attacker could know *all* locations and timestamps that the target victim has visited because it requires non-trivial effort to gather each piece of prior knowledge from so many possible locations at different time. Thus, it is reasonable to assume that the attacker's prior knowledge is bounded by at most L pairs of locations and timestamps that the target victim has visited. A recent work [61] on transaction data anonymization also utilized a similar assumption to bound the prior knowledge of an attacker, but their privacy models together with their methods are not applicable to

anonymizing path data of moving objects.

Based on this assumption, we define a new privacy model called *LKC-privacy* for anonymizing high-dimensional, sparse RFID data. The general intuition is to ensure that every possible subsequence q with maximum length L in any path of a RFID data table T is shared by at least K records in T and the confidence of inferring any sensitive values S from q is not greater than C, where L and K are positive integer thresholds, $0 \leq C \leq 1$ is a real number threshold, and S is a set of sensitive values specified by the data holder. LKC-privacy bounds the probability of a successful record linking attack to be $\leq 1/K$ and bounds the probability of a successful attribute linking attack to be $\leq C$, provided that the attacker's prior knowledge on the target victim is not more than L pairs of locations and timestamps. Table 2 shows an example of anonymous table T' that satisfies $(2, 2, 50\%)$-privacy by suppressing $a1, d2, e4$ from Table 1. Every possible subsequence q with maximum length 2 is shared by at least 2 records and the confidence of inferring the sensitive value HIV from q is not greater than 50%. In contrast, to achieve traditional 2-anonymity, we need to further suppress $b3, c7, e9$, resulting in much higher information loss.

1.4 Research Contribution

RFID mining technique has been introduced by [25]. As a result, research on RFID proliferated quickly, but most work focuses on utilizing RFID [23] [24] and solutions for addressing its privacy issues are limited. A comprehensive privacy-preserving information

system must protect its data throughout its lifecycle, from data collection to data analysis. a majority of previous works on privacy-preserving RFID technology [47] focused on the threats caused by the physical RFID tags and proposed techniques like killing tags which permanently disable the RFID chip, sleeping tags which temporary disable the RFID chip, and EPC re-encryption [30]. They addressed the privacy and security issues at the communication layer among tags and readers, but ignored the protection of the database layer, where a large amount of RFID data actually resides. This thesis provides a complement to the existing privacy-preserving RFID hardware technology.

To the best of our knowledge, this is the first work on anonymizing high-dimensional, sparse RFID data. Our contributions in this thesis are summarized as follows:

- We identify a new privacy problem in RFID data and generalize the requirement to formalize the RFID privacy protection model (Chapter 3).
- We formally define a new privacy model, called LKC-privacy (Chapter 4), for anonymizing high-dimensional, sparse RFID data.
- We propose an efficient anonymization algorithm to transform a table to satisfy a given LKC-privacy requirement without compromising the useful data for analysis.
- We develope a RFID data generator to simulate real-life moving object data.
- We implement the proposed model and evaluate the performance and method in terms of data quality, efficiency, and scalability.

1.5 Thesis Organization

This thesis is organized as follows: we identify the related work in Chapter 2, formally define the privacy model in Chapter 3, present an anonymization algorithm in Chapter 4, experimentally evaluate our proposed method in Chapter 5, and conclude in Chapter 6.

Chapter 2

Related Work

In this chapter, we review the related literature in privacy-preserving data publishing categorized by types of data. Namely, these are relational data (Chapter 2.1), transaction data (Chapter 2.2), and moving object data (Chapter 2.3). We first illustrate the privacy threats caused by publishing different types of data. Then, we discuss the privacy models that thwart the identified privacy threats and the anonymization methods to achieve these privacy models.

2.1 Privacy Models for Relational Data

Relational data is the most common form of store structured data. A relational database consists of a collection of data tables. Each table consists of a set of attributes. Typically, a data table T has the form [16] [27]:

$$T(EI, QID, SI, NSI),$$

where the *Explicit Identifier* (EI) is a set of attributes, such as *Social Insurance Number* (SIN), name and driver license number, consisting of unique information that can identify a record owner. The *Quasi-Identifier* (QID) is a set of attributes that does not contain

explicit identifying information, but some combination of QID values, if specific enough, could potentially identify some record owners. The *Sensitive Information* (SI) is a set of attributes containing personal sensitive information, such as salary and health diagnosis of the record owners and the *Non-Sensitive Information* (NSI) is a set of attributes that does not belong to the above three categories, but is useful for the data publishing purpose. In this thesis, we assume that each record in T represents the information of one record owner, and each record owner has only one record in T. We also assume that SI is important for the purpose of data publishing; otherwise, SI should be removed first.

We assume that an attacker has access to the published table T and has prior knowledge of some target victims' QID values. Privacy threats occur in a table T if an attacker can identify some target victims' records or their sensitive information. We also assume that the EI has been removed before publishing T, but Sweeney [48] showed that even though the EI is removed, it is still possible to identify some target victims' information with prior knowledge of QID. There are two typical types of linking attacks, namely record linking and attribute linking. One way to prevent identifying owners' records is to anonymize the QID so that the record is indistinguishable from others. The anonymized table has the form

$$T(QID', SI, NSI),$$

where QID' is the anonymized version of the QID from the original table T. Below, we use examples to illustrate the latest developments in privacy models and the anonymization methods for achieving the privacy models in relational data.

2.1.1 Record Linking

Let $P(QID)$ be the prior knowledge of an attacker on a target victim. An attacker could use $P(QID)$ to identify a group of records in T that may belong to the victim. The following example illustrates the privacy threat caused by record linking.

Example 2.1.1. A bank wants to release its corporate employees' payroll information, in Table 3, to a third party data mining company. The explicit information, such as name and phone number, can uniquely identify an owner's record and salary information. Thus, $EI = \langle Name, Phone \rangle$ is removed. Table 4 shows the employees' payroll information without EI. Given that an attacker knows that Smith is born in 1975 and works in the DP101 department as prior knowledge, the attacker can uniquely identify the first record in Table 4 to be Smith's record because this is the only record matching the prior knowledge. ■

Table 3: Example of employee payroll table E

Name	Birth Year	Phone	Zip	Gender	Department	Salary
Smith	1975	568-9854	K5G 1Y4	M	DP101	50,000
Mac	1976	589-9556	K5G 1Y4	M	DP101	50,000
Hack	1968	658-9875	L9A 2B1	F	DP102	230,000
Simpson	1968	449-9896	L9A 2B1	F	DP102	60,000
Grandson	1969	589-8546	H3A 2B1	M	DP103	70,000
Chanson	1968	984-3204	H3A 2B1	M	DP103	190,000

Table 4: Example of employee payroll table E without EI

Birth Year	Zip	Gender	Department	Salary
1975	K5G 1Y4	M	DP101	50,000
1976	K5G 1Y4	M	DP101	50,000
1968	L9A 2B1	F	DP102	230,000
1968	L9A 2B1	F	DP102	60,000
1969	H3A 2B1	M	DP103	70,000
1968	H3A 2B1	M	DP103	190,000

Many data masking schema have been proposed to thwart the privacy threat caused by record linking. Perturbation is a widely used approach in statistical control for this purpose. The general idea is to add noise to numerical data [2] [9], such as age and salary, while preserving the statistical mean, correlation, or some properties in the data for the purpose of data mining [5] [15] [32] [6] [12] [14]. Kargupta et al. [31] showed that it is possible to

recover the real data from the perturbed data if the distribution of noise is known. Huang et al. [28] proposed an improved method to randomize the data to avoid this problem.

Samarati and Sweeney [46] [45] proposed an alternative privacy model, called K-anonymity, to thwart the privacy threat caused by record linking. The intuition is to require that, for each record in the data table T, there exists at least K-1 other records that share the same QID values in T. This privacy model guarantees that the probability of a successful record linking is $\leq 1/K$.

Generalization is a commonly used masking scheme for achieving K-anonymity. A generalization replaces some specific data value with a less specific parent data value based on a user-defined attribute taxonomy tree, e.g., Figure 2. As the value is generalized to more abstract levels, it is more likely to be shared by more records, therefore reducing the chance of linking a record to a record owner.

Example 2.1.2. Table 5 shows a 2-anonymous table on $QID = \langle BirthYear, Zip, Gender, Department \rangle$, meaning that every combination of values on QID in the table is shared by at least two data records. This 2-anonymity is achieved by generalizing 1968 and 1969 to 1960s on the $BirthYear$ in Table 4 based on the user-defined taxonomy tree in Figure 2. The user-defined taxonomy tree is defined for the purposes of capturing the user's domain knowledge and is required before the generalization is performed. 1960s is the parent value of 1968 and 1969, so the value 1960s covers more records in the data table, therefore, reducing the chance of linking the victim's record. Given that the table is 2-anonymous, the maximum probability of a successful record linking is $\leq 50\%$. ■

The database community has spent lots of effort on privacy-preserving data publishing, where the goal is to transform a relational data into an anonymous version for preventing record and attribute linkings. Traditional K-anonymity [7] [33] [45] and its extensions [19] [20] [36] [38] [42] [55] [56] [58] [59] are not applicable to anonymize RFID data due to the curse of high-dimensionality [3] and data sparseness discussed in Chapter 1. We tackle

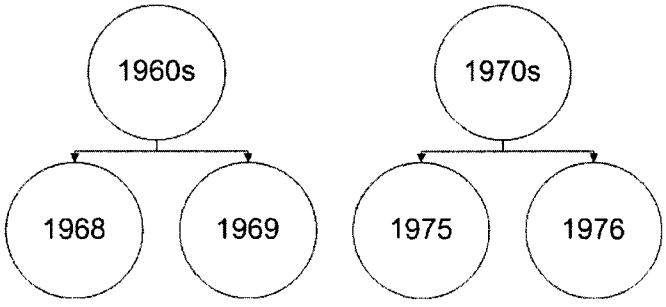


Figure 2: Taxonomy Tree for $BirthYear$

this challenge by exploiting the assumption that the attacker knows at most L pairs of previously visited locations and timestamps by a target victim.

Table 5: Anonymous table E' employee payroll information without EI

Birth Year	Zip	Gender	Department	Salary
1970s	K5G 1Y4	M	DP101	50,000
1970s	K5G 1Y4	M	DP101	50,000
1960s	L9A 2B1	F	DP102	230,000
1960s	L9A 2B1	F	DP102	60,000
1960s	H3A 2B1	M	DP103	70,000
1960s	H3A 2B1	M	DP103	190,000

2.1.2 Attribute Linking

In attribute linking, even though the attacker cannot identify the record of a target victim, the attacker may infer the victim's sensitive information based on the QID if some records in the same QID groups share the same sensitive information. This type of attribute linking is possible even though the table has been K-anonymized. A naive solution is to simply

remove the sensitive information. Yet, if the sensitive information is important for data publishing purposes, such a solution is not applicable. Wang et al. [54] [56] proposed a model to bound the confidence of inferring sensitive information. The following example illustrates the intuition.

Example 2.1.3. Consider Table 5. Suppose the attacker knows that Smith's QID is $\langle 1970s,$ $K5G\ 1Y4, M, DP101\rangle$. Due to the first two matching records (record 1 and record 2) share the same QID value, the attacker can infer that $\langle 1970s, K5G\ 1Y4, M, DP101\rangle \rightarrow$ $\$50,000$ with 100% confidence. ■

Wang et al. [54] [56] proposed a privacy model that bounds the confidence of inferring sensitive information from a QID. The privacy requirement is specified in a template, $\langle QID \rightarrow s, C\rangle$, where s is a user-specified sensitive value and C is a user-defined threshold. $Conf\langle QID \rightarrow s\rangle$ denotes the maximum confidence to infer the sensitive value s from any QID values. Thus, a table satisfies $\langle QID \rightarrow s, C\rangle$ only if the confidence of inferring s from any QID value is below or equal to C. For example, assuming the privacy requirement for Table 5 is $K = 2$ and $C = 50\%$, it still violates the privacy requirement even it satisfied $K = 2$ because the confidence of inferring $Salary = \$50,000$ from $\langle 1970s, K5G\ 1Y4, M, DP101\rangle$ is 100% which is $\geq 50\%$.

To prevent attribute linking, Machanavajjhala et al. [38] [39] proposed a privacy model called the ℓ-diversity, which requires every QID group to have at least ℓ distinct sensitive information. Therefore, a larger ℓ implies less chances of inferring a particular sensitive information in a QID group. Compared to confidence bounding, ℓ-diversity does not quantify the probability value. In addition, it is difficult to define different protection levels with different sensitive information groups. Machanavajjhala et al. [38] [39] proposed two other models called positive disclosure-recursive (c, ℓ)-diversity and negative/-positive disclosure-recursive(c, ℓ) diversity, which can better model the attacker's background knowledge. These notions serve similar purposes for the privacy template proposed

in [54] [56].

Wong et al. [58] presented an integrated model called (α, K)-anonymity. (α, K)-anonymity requires that every qid group to contain at least K records and $conf\langle qid \rightarrow s\rangle \leq \alpha$ for any sensitive value s, where K and α are user-defined thresholds.

Zhang et al. [63] proposed the (k, e)-anonymity model focusing on eliminating attribute linking on numerical sensitive attributes such as salary, while previous models focused on the categorical sensitive attribute such as health diagnosis. (k, e)-anonymity requires that each QID group must have at least k sensitive information with a range of e. One limitation of (k, e)-anonymity is that if some sensitive values occur frequently within a subrange of d, then the attacker could still confidently infer the subrange in a group.

Recall that the ℓ-diversity privacy model requires every QID to have at least ℓ distinct sensitive values. Li et al. [35] found that when the overall distribution of a sensitive attribute is skewed, ℓ-diversity does not prevent attribute linking attacks. Consider the payroll information in Table 4 where only 20% of people have salary of $120,000. Suppose that there is a 2-diversity QID group where 50% of people have a salary of $50,000 and 50% of people have a salary of $120,000. This group presents a serious privacy threat because any record owner in the group could be inferred as having $120,000 with 50% confidence, compared to 20% in the overall table. Li et al. [35] proposed a privacy model, called t-closeness, to require the distribution of a sensitive attribute in any group on QID to be close in the overall table and the closeness is within t. Li et al. [34] further extended the work to apply to the numerical sensitive attribute. Applying t-closeness will greatly damage the data usefulness because it requires the sensitive distribution to be the same among the entire group. Domingo-Ferrer and Torra [11] adjusted the thresholds to increase the risk of skewness and thus reduce the data utility loss. In our thesis, we use confidence bounding to restrict the probability for attackers to infer the sensitive information and it is easy to adjust the trade-off between privacy protection and information utility.

Bu et al. [10] presented a privacy-preserving method for serial data publishing with permanent sensitive values and dynamic registration lists. The authors assumed that the attacker had limited participation knowledge and bounded the probability of linking between any individual and any sensitive information with a threshold $1/\ell$ and mixed the sensitive information with NSI.

Fung et al. [19] [20] presented a top-down specialization (TDS) method to generalizes a table by specializing it from the general state to satisfy k-anonymization. TDS is able to handle categorical attributes and numerical attributes. Fung et al. [17] [18] further extended the k-anonymization algorithm to preserve the information for clustering analysis. They first partitioned the original data into clusters and then apply TDS to meet the k-anonymization. Mohammed et al. [43] and Trojer et al. [53] extended the idea to distributed data mashup application.

The focus of this thesis is to present a privacy-preserving data publishing method for high-dimensional, sequential data which is fundamentally different from the relational data discussed in this section.

2.2 Privacy Model for Transaction Data

Recent study has shown that publishing transaction data may also pose a privacy threat on sensitive information linking. A transaction database [27] is a table where each record contains a set of items. Examples of transaction data are credit card transaction data, medical notes and e-mails. A transaction typically contains a transaction number and a set of associated items. In real-life transaction database, the transaction table may be linked to other relational database, which contains more information describing the transaction. The information is usually rich and valuable to data mining. A full report of transaction data may describe full image of personal activities and potentially person's sensitive information.

The following case study illustrates the privacy threat caused by publishing transaction

data. AOL released a search query database without record owners' name [37]. Afterwards, Record No. 4417749 was traced back to Ms. Thelma Arnold. Even the name and user's address have already been removed from the database, attackers can still identify the record owner by combining the published data query.

Traditional privacy models K-anonymity and ℓ-diversity aims at limiting the linking via the QID such as Birth Year, Zip, Gender and Department. In transaction data, each item may be considered as an attribute in the QID and transaction data is often high-dimensional. For example, in a credit card data log, there may exist thousands to millions of distinct items. Each transaction contains only a very small fraction of items. Thus, transaction data is often high-dimensional and sparse. Due to the curse of high-dimensionality [3], applying traditional privacy models often results in suppressing most of the items, making the published data useless. Aggarwal et al. [3] discussed the problem of the high-dimensionality on K-anonymity but did not provide a solution to the problem. In this thesis, we present a new privacy model for anonymizing high-dimensional data.

Recently, there were some studies on anonymizing high-dimensional transaction data [22] [50] [61] [60]. Ghinita et al. [22] proposed a permutation method in which the general idea is to first group transactions with close proximity and then associate each group to a set of mixed sensitive values. In our model, the attacker's prior knowledge $\langle a, b\rangle$ is considered to be different from prior knowledge $\langle b, a\rangle$; therefore, the proposed privacy models and anonymization methods by [22] [50] [61] for anonymizing transaction data is not applicable to our problem. Terrovotis et al. [50] and Su et al. [61] extended the traditional K-anonymity model by assuming that the attacker knows that at most m transaction items of the target victims. All these works [22] [50] [61] considered a transaction as a *set* of items. Terrovitis et al. [51] proposed a K_M model to restrict the attacker's prior knowledge to M items in a transaction record. Their works are different than our thesis. First, their transaction records have no time sequence. Second, their transaction database has no

sensitive attributes. Finally, they used generalization techniques to achieve K-anonymity. Aggarwal et al. [4] proposed a sketch based model by using perturbation schema in a way to retain the original data utility after anonymization.

2.2.1 Record Linking

Let $P(QID)$ be the prior knowledge of an attacker on a target victim. The attacker may use $P(QID)$ to identify the target victim's record. The following example illustrates this privacy threat of record linking in the context of transaction data.

Example 2.2.1. Consider a credit card company that would like to release its customer transaction data at Table 6 for data mining purposes. TID stands for Transaction ID. A record owner has purchased items such as pencils, coffee, and medicine, denoted by $\{a, b, c, d\}$. Medical history is the victim's health diagnosis associated with each record. Assuming our pre-defined privacy requirement is K = 2, Table 6 violates the privacy requirement. For example, if an attacker knows that the victim has purchased items $\{a, b\}$, among the entire records, only T2 contains $\{a, b\}$. Thus, the attacker can infer that T2 is the victim's record, together with her other purchased items and medical history. ■

Table 6: Credit card item transaction table C [61]

TID	Purchased Items	Medical History
T1	a, c, d, f, g	Diabetes
T2	a, b, c, f	Hepatitis
T3	b, d, f, x	Hepatitis
T4	b, c, g, y, z	HIV
T5	a, c, f, g	HIV

2.2.2 Attribute Linking

Due to the sparseness of transaction data, there are not many records sharing the same QID values; therefore, with the attacker's prior knowledge $P(QID)$, the attacker can easily infer victim's sensitive information. For example, if an attacker knows that a victim has purchased $\{b, f\}$, record T2 and T5 satisfy our K-anonymity requirements but attacker can still infer that the victims has Hepatitis.

In chapter 2.1, we have discussed using generalization to achieve K-anonymity. Here, we study another masking scheme called suppression. Suppression replaces some information with null information. There are two schema of the suppression: global suppression and local suppression. Global suppression removes *all* the instances of a given value from the entire data set. Local suppression removes *some* of the instances of a given value from the data set with the goal of minimizing loss.

Applying the traditional K-anonymity and confidence bounding [56] models on transaction data would result in suppressing lots of data, making the published data useless for data analysis. Xu et al. [61] assumed that attackers only have h pieces of prior knowledge in QID. This assumption can significantly reduce the information loss in the anonymous data.

Example 2.2.2. Consider Table 6, to achieve the privacy requirements with $K = 2$, $h = 2$ and $C = 80\%$, we can globally suppress $\langle x, y, z \rangle$. Table 7 shows the resulting table that satisfies the privacy requirement. ■

2.3 Privacy Model for Moving Object Data

Moving object data is a special type of transaction data. Compared to transaction data, the items in a moving object data are often time and location dependent. Location-based

Table 7: Anonymous credit card item transaction table C' [61]

TID	Purchase Item	Medical History
T1	a, c, d, f, g	Diabetes
T2	a, b, c, f	Hepatitis
T3	b, d, f	Hepatitis
T4	b, c, g	HIV
T5	a, c, f, g	HIV

services (LBS) generate large amounts of moving object data based on their physical location and time. Examples of the moving object data are mobile phone data, GPS Data, Web Query Data and RFID data. These data provide abundant data sources for mining researchers to predict market trends, human purchase patterns and traffic patterns to improve current technology to better serve society. Even though the LBS data are valuable and useful, research [13] has shown that 24% of users are concerned about their privacy caused by the location-based services data. Moving object data by nature are time dependent, location dependent, high-dimensional and sparse. It thus poses a challenge to protect the data in a way to that preserves the useful information.

Various routing and messaging techniques have been proposed to protect the privacy of subscribers in a mobile network. The Mix Zones System [8] provided anonymous messaging services by delaying and reordering messages from subscribers within mix zones to confuse an attacker. Other mechanisms, such as cloaking [26] and location-based K-anonymity [21], concealed a subscriber within a group of K subscribers. A subscriber is considered K-anonymous if she is indistinguishable from at least $K-1$ other subscribers. This privacy requirement is achieved by generalizing the disclosed locations and timestamps of the messages, or by delaying the messages. Such anonymous messaging techniques are not applicable to RFID data because RFID data, such as journey data, consists of *sequences* of locations with different timestamps, rather than simply the location of the sender and the receiver. Mix Zones is mainly designed for anonymizing dynamic messages

for location-based services (LBS). It is very different from releasing a large set of location-based RFID data. Papadimitriou et al. [44] presented a privacy issue on publishing time series data and proposed a perturbation method to add some "similar information" to the original data with the goal of deviating the linking attacks.

Currently, there are few works [1] [49] [62] on anonymizing moving object and extend the traditional K-anonymity model to anonymize a set of moving objects. [1] proposed a method to ensure at least K moving object are within its radius, where the radius is a user-specified threshold. The privacy requirements can be achieved by space translation and adding noise to original paths. Our approach is different from [1] in two major aspects. First, their model does not consider the privacy threat caused by attribute linking between the path and the sensitive attribute. Second, they assume that all moving objects have continuous timestamps. This assumption may hold in mobile phone or LBS applications, where the user's location is continuously detected while the phone is turned on. However, this assumption does not hold for RFID because a RFID-tagged object (e.g., smart cards used in transportation) is unlikely to be continuously detected by a RFID reader. These differences imply different privacy threats and models. Terrovitis et al. [49] assumed a very different attack model on moving objects. They considered that the locations themselves are sensitive information and the attacker attempts to infer some sensitive locations visited by the target victims are unknown to the attacker. They did not specifically address the high-dimensionality problem in RFID data, which is the theme of this thesis. Malin et al. [40] also studied the privacy threats in location-based data conducted in hospitals.

Chapter 3

Problem Definition

In this chapter, we formally define the anonymization problem for RFID data. Section 3.1 defines the format of an object-specific path table. Section 3.2 formally defines the privacy threats and the background knowledge of an attacker. Section 3.3 formally defines our proposed privacy model, namely LKC-Privacy, followed by a problem statement in Section 3.4.

3.1 Object-Specific Path Table

A typical RFID system generates a sequence of RFID data records of the form $\langle EPC, loc, t\rangle$, where each record indicates a RFID reader in location loc has detected an object having electronic product code (EPC) at time t. We assume that the RFID-tagged item is attached to or carried by some moving object, for example, patients in a hospital or patients in a public transit system.

A *pair* $(loc_i t_i)$ represents that the object has visited location loc_i at time t_i. The *path* of an object, denoted by $\langle (loc_1 t_1) \dots (loc_n t_n)\rangle$, is a sequence of pairs that can be obtained by first grouping the RFID records by EPC and then sorting the records in each group by timestamps. A timestamp is the entry time to a location, so the object is assumed to

be staying in the same location until its new location is detected by another reader. An object may revisit the same locations at different timestamps, but consecutive pairs having the same location are duplicates and, therefore, are removed. For example, in $\langle a1 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow c7 \rightarrow b8\rangle$, $b4$ and $b6$ are removed but $b8$ is kept. At any time, an object can be at only one location, so $a1 \rightarrow b1$ is not a valid sequence. Timestamps in a path must increase monotonically.

An *object-specific path table* T is a collection of records in the form

$$\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n)\rangle : s_1, \ldots, s_p : d_1, \ldots, d_m,$$

where $\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n)\rangle$ is a path, $s_i \in S_i$ are sensitive attributes, and $d_i \in D_i$ are quasi-identifying (QID) attributes associated with the object. In the rest of this thesis, the term "record" refers to the above form. The QID attributes are relational data and can be anonymized by existing methods [20] [36] [38] [45] [56] for relational data. This thesis focuses on the paths and sensitive attributes. Table 1 gives an example of object-specific path table.

3.2 Privacy Threats

Suppose a data holder wants to publish an object-specific path table T to some recipient(s) for data analysis. Explicit identifiers, e.g., name, SSN, and EPC, have been removed. The paths, together with the object-specific attributes, are assumed to be important for the task of data analysis; otherwise, they could be removed. One recipient, the attacker, seeks to identify the record or sensitive values of some target victim V in T. As explained in Chapter 1, we assume that the attacker knows at most L pairs of location and timestamp that the victim V has previously visited. We use $\kappa = \langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_z t_z)\rangle$ to denote such prior knowledge, where $z \leq L$. Using the prior knowledge κ, the attacker could identify a group of records in T, denoted by $G(\kappa)$, that "matches" κ. A record *matches* κ if κ is a

disjoint subsequence of the path in the record. For example in Table 1, if $\kappa = \langle e4 \rightarrow c7 \rangle$, then $EPC\#1$ $[\langle a1 \rightarrow d2 \rightarrow b3 \rightarrow e4 \rightarrow f6 \rightarrow c7 \rangle : HIV]$ matches κ, but $EPC\#4$ $[\langle d2 \rightarrow f6 \rightarrow c7 \rightarrow e8 \rangle : Allergy]$ does not.

The notion of *matching* is formally defined as following.

Definition 3.2.1 (Matching). A pair $(loc_i t_i)$ *matches* a pair $(loc_j t_j)$ if $loc_i = loc_j$ and $t_i = t_j$. A path p_x *covers* a path p_y if, for every pair $(loc_y t_y)$ in p_y, there exists a pair $(loc_x t_x)$ in p_x that matches $(loc_y t_y)$. A record *matches* κ if the path of the record covers κ. ■

An attacker could utilize $G(\kappa)$ to perform two types of privacy attacks:

1. *Record linking*: $G(\kappa)$ is a set of candidate records that contains the victim V's record. If the group size of $G(\kappa)$, denoted by $|G(\kappa)|$, is small, then the attacker may identify V's record from $G(\kappa)$, and therefore, V's sensitive value.

2. *Attribute linking*: Given $G(\kappa)$, the attacker may infer that V has sensitive value s with confidence $Conf(s|G(\kappa)) = \frac{|G(\kappa \bigcup s)|}{|G(\kappa)|}$, where $G(\kappa \bigcup s)$ denotes the set of records containing both κ and s. $Conf(s|G(\kappa))$ is the percentage of the records in $G(\kappa)$ containing s. The privacy of V is at risk if $Conf(s|G(\kappa))$ is high.

Example 1.2.1 illustrates these two types of attacks.

3.3 Privacy Models

The problem studied in this thesis is to transform the raw object-specific path table T to a version T' that is immunized against record and attribute linking. We define two separate privacy models *LK-anonymity* and *LC-dilution* to thwart record linking and attribute linking, respectively, followed by a unified model. The attacker's prior knowledge κ could be any subsequence q with a maximum length L of any path in T.

3.3.1 LK-Anonymity

Definition 3.3.1 (LK-anonymity). An object-specific path table T *satisfies* LK*-anonymity* if and only if $|G(q)| \geq K$ for any subsequence q with $|q| \leq L$ of any path in T, where K is a positive anonymity threshold. ■

3.3.2 LC-Dilution

Definition 3.3.2 (LC-dilution). Let S be a set of data holder-specified sensitive values from sensitive attributes $S_1, \ldots, S_m$. An object-specific path table T *satisfies* LC*-dilution* if and only if $Conf(s|G(q)) \leq C$ for any $s \in S$ and for any subsequence q with $|q| \leq L$ of any path in T, where $0 \leq C \leq 1$ is a confidence threshold. ■

3.3.3 LKC-Privacy

Definition 3.3.3 (LKC-privacy). An object-specific path table T *satisfies* LKC*-privacy* if T satisfies both LK-anonymity and LC-dilution. ■

LK-anonymity bounds the probability of a successful record linking to $\leq 1/K$. LC-dilution bounds the probability of a successful attribute linking to $\leq C$. LKC-privacy bounds both. Note, not all values in sensitive attributes $S_1, \ldots, S_m$ are sensitive. For example, HIV could be sensitive, but flu may not be. Our proposed privacy model is flexible to accommodate different privacy need by allowing the data holder to specify a set of sensitive values S in Definition 3.3.2.

3.4 Problem Statement

We can transform an object-specific path table T to satisfy LKC-privacy by performing a sequence of suppressions on selected pairs from T. In this thesis, we employ *global suppression*, meaning that if a pair p is chosen to be suppressed, all instances of p in T are

suppressed. We use Sup to denote the set of suppressed pairs. Table 2 is the result of suppressing $a1$, $d2$, and $e4$ from Table 1. This suppression scheme offers several advantages over generalization for anonymizing RFID data. First, it does not require a predefined taxonomy tree for generalization, which often is unavailable in real-life databases. Second, RFID data can be extremely sparse. Enforcing generalization on RFID data may result in generalizing many "neighbor" objects even though if there is only a small number of outlier pairs, such as $a1$ in Table 1. In addition, generalization matching schema is computationly expensive. Furthermore, generalization requires user to define a taxonomy tree which may affect the accuracy of the data mining analysis. Suppression offers the flexibility of removing those outliers without affecting the rest of the data.

Definition 3.4.1 (Anonymization for RFID). Given an object-specific path table T a LKC-privacy requirement, and a set of sensitive values S, the problem of *anonymization for RFID* is to identify a transformed version T' that satisfies the LKC-privacy requirement by suppressing a minimal number of instances of pairs in T. ■

K-anonymity [45] is a special case of LKC-privacy with $L = \infty$ and $C = 100\%$. Confidence bounding [56] is a special case LKC-privacy with $L = \infty$ and $K = 1$. Given that achieving optimal K-anonymity and optimal confidence bounding have been proven to be NP-hard [41] [56], achieving optimal LKC-privacy is also NP-hard. Thus, we propose a greedy algorithm to efficiently identify a sub-optimal solution. LKC-privacy has also been proposed by anonymizing relational healthcare data [42].

Chapter 4

Anonymization Method

Given an object-specific path table T and a LKC-privacy requirement, our goal is to remove all "violations" from T, where a *violation* is a subsequence of a path in T that violates the LKC-privacy requirement. We first present an efficient algorithm for identifying *all* violations in Chapter 4.1, followed by a greedy algorithm to remove all violations in Chapter 4.2.

4.1 Identifying Violations

A subsequence q in T is a *violation* if its length is less than the maximum length threshold L and its group $G(q)$ violates LK-anonymity, LC-dilution, or both. The adversary's prior knowledge κ could be any of such subsequence q. Thus, removing all violations means eliminating all possible channels of record and attribute linking attacks.

Definition 4.1.1 (Violation). Let q be a subsequence of a path in T with $|q| \leq L$ and $|G(q)| > 0$. q is a *violation* with respect to a LKC-privacy requirement if $|G(q)| < K$ or $Conf(s|G(q)) > C$. ■

Example 4.1.1. In Table 1, a sequence $q_1 = \langle e4 \rightarrow c7 \rangle$ is a violation if $K = 2$ because

$|G(q_1)| = 1 < 2$. A sequence $q_2 = \langle d2 \rightarrow f6 \rangle$ is a violation if $C = 50\%$ and $S = \{HIV\}$ because $Conf(HIV|G(q_2)) = 67\% > 50\%$. ■

We note two properties in the notion of violation. (1) If q is a violation with $|G(q)| < K$, then any super sequence of q, denoted by q', is also a violation because $|G(q')| \leq |G(q)| < K$. This property has two implications. First, it implies that the number of violations could be huge, so it is not feasible to first generate all violations and then remove them. Second, if $L \leq L'$, a table T satisfying $L'K$-anonymity must satisfy LK-anonymity because $|G(q)| \geq |G(q')| \geq K$. (2) If q is a violation with $Conf(s|G(q)) > C$ and $|G(q)| \geq K$, its super sequence q' may or may not be a violation because $Conf(s|G(q')) \geq Conf(s|G(q))$ does not always hold. Thus, to achieve LC-dilution, it is insufficient to ensure any subsequence q with length L in T to satisfy $Conf(s|G(q)) \geq C$. Instead, we need to ensure any subsequence q with length less than or equal to L in T to satisfy $Conf(s|G(q)) \geq C$.

Enumerating all possible violations is infeasible. Our insight is that among all the violations, there exists some minimal sequences called "critical violations". We show that a violation exists in table T if and only if a critical violation exists in T.

4.1.1 Critical Violation Tree

Definition 4.1.2 (Critical violation)**.** A violation q is a *critical violation* if every proper subsequence of q is a non-violation. ■

Example 4.1.2. In Table 1, if $K = 2$, $C = 50\%$, $S = \{HIV\}$, a sequence $q_1 = \langle e4 \rightarrow c7 \rangle$ is a critical violation because $|G(q_1)| = 1 < 2$, and both $\langle e4 \rangle$ and $\langle c7 \rangle$ are non-violations. A sequence $q_2 = \langle d2 \rightarrow e4 \rightarrow c7 \rangle$ is a violation but it is a not a critical violation because its subsequence $\langle e4 \rightarrow c7 \rangle$ is a violation. ■

Observation 4.1.1. A table T' satisfies LKC-privacy if and only if T' contains no critical violation because each violation contains a critical violation. Thus, if T' contains no critical violations, then T' contains no violations. ■

4.1.2 Efficient Tree Scan Algorithm

Next, we propose an algorithm to efficiently identify all critical violations in T with respect to a LKC-privacy requirement. Based on Definition 4.1.2, we generate all critical violations of size $i+1$, denoted by V_{i+1}, by incrementally extending non-violations of size i, denoted by U_i, with an additional pair.

Procedure 1 Generate Critical Violations (GenViolations)

Input: Raw RFID path table T
Input: Thresholds L, K, and C.
Input: Sensitive values S.
Output: Critical violations V.

```
1:  let candidate set C_1 be the set of all distinct pairs in T;
2:  i = 1;
3:  repeat
4:     scan T once to obtain |G(q)| and Conf(s|G(q)) for every sequence q ∈ C_i and for every
       sensitive value s ∈ S;
5:     for all sequence q ∈ C_i do
6:        if |G(q)| > 0 then
7:           if |G(q)| < K or Conf(s|G(q)) > C for any s ∈ S then
8:              add q to V_i;
9:           else
10:             add q to U_i;
11:          end if
12:       end if
13:    end for
14:    ++i;
15:    generate candidate set C_i by U_{i-1} ⋈ U_{i-1};
16:    for all sequence q ∈ C_i do
17:       if q is a super sequence of v for any v ∈ V_{i-1} then
18:          remove q from C_i;
19:       end if
20:    end for
21: until i > L or C_i = ∅
22: return V = V_1 ⋃ ... ⋃ V_{i-1};
```

Procedure 1 summarizes the steps for generating critical violations. Line 1 initializes the candidate set C_1 to be the set of all distinct pairs in any paths in the raw table T. Line 4 scans the raw data once to obtain the support counts to compute $|G(q)|$ and $Conf(s|G(q))$ for every sequence $q \in C_i$ and for every sensitive value $s \in S$. Lines 5-13 loops through

every candidate $q \in C_i$ of $|G(q)| > 0$, and puts q to the critical violation set V_i if it violates LK-anonymity or LC-dilution; otherwise, puts q to the non-violation set U_i. Once a violation is found, we remove it from subsequent iterations because its super sequence must not be a critical violation. Line 15 generates a candidate set C_i by self-joining U_{i-1}. Two sequences $q_x = \langle (loc_1^x t_1^x) \rightarrow \ldots \rightarrow (loc_{i-1}^x t_{i-1}^x) \rangle$ and $q_y = \langle (loc_1^y t_1^y) \rightarrow \ldots \rightarrow (loc_{i-1}^y t_{i-1}^y) \rangle$ in U_{i-1} can be joined only if the first $i-2$ pairs of q_x and q_y are identical and $t_{i-1}^x < t_{i-1}^y$. The joined sequence is $\langle (loc_1^x t_1^x) \rightarrow \ldots \rightarrow (loc_{i-1}^x t_{i-1}^x) \rightarrow (loc_{i-1}^y t_{i-1}^y) \rangle$. Lines 16-20 removes a candidate q from C_i if q is a super sequence of any sequence in V_{i-1} because all proper subsequences of a critical violation must be a non-violation.

Example 4.1.3. Consider Table 1 with $L = 2$, $K = 2$, $C = 50\%$, and $S = \{HIV\}$. First, we generate candidate set $C_1 = \{a1, d2, b3, e4, c5, f6, c7, e8, e9\}$, which is a set of distinct pairs in T. Then, we scan Table 1 to identify the critical violations from C_1 and put them in $V_1 = \{a1\}$. The remaining sequences are non-violations $U_1 = \{d2, b3, e4, c5, f6, c7, e8, e9\}$. Next, we generate $C_2 = \{d2b3, d2e4, d2c5, d2f6, d2c7, d2e8, d2e9, b3e4, b3c5, b3f6, b3c7, b3e8, b3e9, e4c5, e4f6, e4c7, e4e8, e4e9, c5f6, c5c7, c5e8, c5e9, f6c7, f6e8, f6e9, c7e8, c7e9, e8e9\}$ and scan once Table 1 to determine critical violations $V_2 = \{d2b3, d2e4, d2f6, d2e8, d2e9, e4c7, e4e8\}$.

4.2 Removing Violations

4.2.1 Selection Score Function

We propose a greedy algorithm to transform raw table T to an anonymous table T' with respect to a given LKC-privacy requirement by a sequence of suppressions. In each iteration, the algorithm selects a suppression on value v based on a greedy selection function. In general, a suppression on a value v in T increases privacy because it removes critical violations, and decreases information utility because it suppresses pairs in T. Therefore,

we define the greedy function, $Score(p)$, to select a suppression on a pair p that maximizes the number of critical violations removed and minimizes the number of pair instances suppressed in T. $Score(p)$ is formally defined as follows:

$$Score(p) = \frac{PrivGain(p)}{InfoLoss(p)}, \tag{1}$$

where $PrivGain(p)$ is the number of critical violations containing pair p and $InfoLoss(p)$ is the number of instances of pair p in T. Alternative greedy functions could be

$$Score(p) = PrivGain(p), \tag{2}$$

which aims at eliminating all critical violations but ignores the information loss caused by the suppression, or

$$Score(p) = \frac{1}{InfoLoss(p)}, \tag{3}$$

which aims at minimizing the number of suppressed instances in T but ignores how many critical violations can be removed by the suppression. In Chapter 5, we will evaluate the performance of these variations.

4.2.2 Monotonicity Analysis

For any subsequence p of path in T, $G(p)$ monotonically increases with respect to a global suppression. K-Anonymity and ℓ-diversity [38] satisfy the *monotonicity property*. In our case, when T preserves the privacy, each suppression will also preserve privacy. The count for each pairs in the table would not decrease after suppression. The maximum probability to infer a sensitive information would not increase after suppression; therefore, our LKC-privacy algorithm satisfies the *monotonicity property*. ■

Algorithm 2 RFID Data Anonymizer

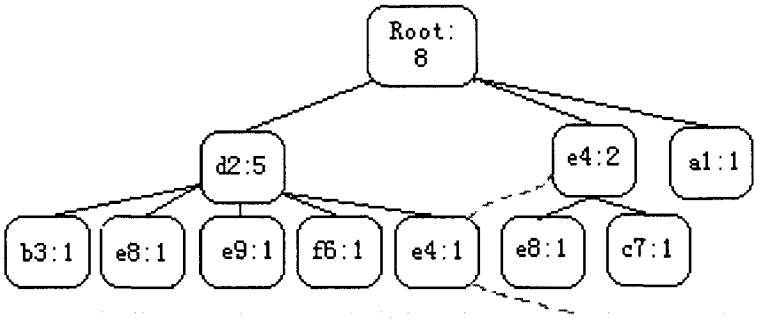
Input: Raw RFID path table T
Input: Thresholds L, K, and C.
Input: Sensitive values S.
Output: Anonymous T' that satisfies LKC-privacy.

1: V = Call GenViolations(T, L, K, C, S) in Procedure 1;
2: build the Critical Violation Tree (CVT) with Score Table;
3: **while** Score Table is not empty **do**
4: select winner pair w that has the highest $Score$;
5: delete all critical violations containing w in CVT;
6: update $Score$ of a candidate x if both w and x were contained in the same critical violation;
7: remove w in Score Table;
8: add w to Sup;
9: **end while**
10: for every $w \in Sup$, suppress all instances of w from T;
11: **return** the suppressed T as T';

4.2.3 Greedy Algorithm

Algorithm 2 summarizes the RFID data anonymization algorithm. Lines 1-2 call Procedure 1 to generate all critical violations and build a tree to represent them. At each iteration in Lines 3-9, the algorithm selects the winner pair w that has the highest $Score(p)$ among all candidates for suppression, removes the critical violations containing w, and incrementally updates the $Score$ of the affected candidates due to the suppression on w. Sup denotes the set of all suppressed winner pairs. They are collectively suppressed in Line 10 in one scan of T. Finally, Algorithm 2 returns the anonymized T as T'. The most expensive operations are to identify the critical violations containing w and to update the $Score$ of the affected candidates. Below, we propose a data structure called *critical violation tree (CVT)* to efficiently support these operations.

Definition 4.2.1 (Critical Violation Tree (CVT)). CVT is a tree structure that represents each critical violation as a tree path from root-to-leaf. Each node keeps track of a count of critical violations sharing the same prefix. The count at the root is the total number of critical violations. CVT has a Score Table that maintains every candidate pair p for

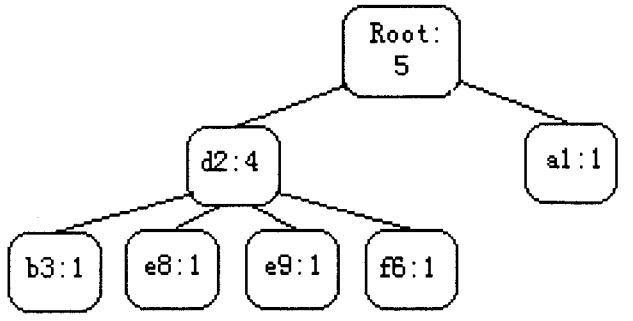


Pair	d2	e8	b3	e9	f6	e4	c7	a1
PrivGain	5	2	1	1	1	3	1	1
InfoLoss	4	3	3	3	6	2	5	1
Score	1.3	0.7	0.3	0.3	0.2	1.5	0.2	1

Figure 3: Initial Critical Violation Tree (CVT)

suppression, together with its $PrivGain(p)$, $InfoLoss(p)$, and $Score(p)$. Each candidate pair p in the Score Table has a link, denoted by $Link_p$, that links up all the nodes in CVT containing p. $PrivGain(p)$ is the sum of the counts of critical violations on $Link_p$. ■

Figure 3 depicts the initial CVT generated from V_1 and V_2 in Example 4.1.3. The winner pair $e4$, which has the highest $Score$, is identified from the Score Table. Then, the algorithm traverses $Link_{e4}$ to identify all critical violations containing $e4$ and deletes them from CVT accordingly. When a winner pair w is suppressed from CVT, the entire branch of w is trimmed. This provides an efficient method for removing critical violations. In Figures 3 and 4, when $e4$ is suppressed, all its descendants are removed as well. The count of critical violations of $e4$'s ancestor nodes is decremented by the count of critical violations of the deleted $e4$ node. If a candidate pair p and the winner pair w are contained in some critical violation, then $PrivGain(p)$, and therefore $Score(p)$, has to be updated for adding up the counts on $Link_p$. For example, after $e4$ is suppressed, $PrivGain(d2)$, $PrivGain(c7)$, and $PrivGain(e8)$ have to be updated. A pair p with $PrivGain(p) = 0$ in Score Table is removed.



Pair	d2	e8	b3	e9	f6	a1
PrivGain	4	1	1	1	1	1
InfoLoss	4	3	3	3	6	1
Score	1	0.3	0.3	0.3	0.2	1

Figure 4: CVT after suppressing $e4$

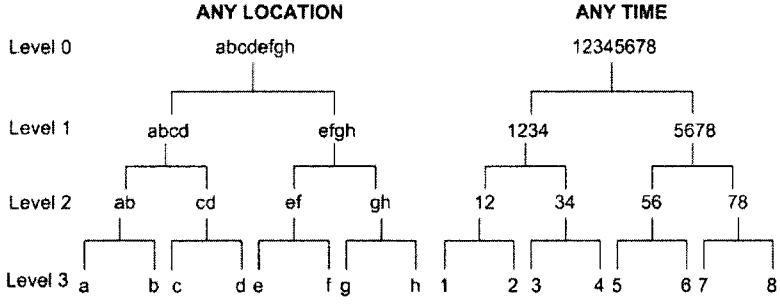


Figure 5: Taxonomy Trees for *Locations* and *Times*

Table 8: Patient-specific path table P

EPC	Path	Diagnosis	...
1	$\langle d4 \rightarrow f7 \rightarrow h8\rangle$	Diabetes	
2	$\langle a1 \rightarrow c2 \rightarrow d4 \rightarrow f5 \rightarrow h8\rangle$	HIV	
3	$\langle d4 \rightarrow e5 \rightarrow f7 \rightarrow h8\rangle$	Flu	
4	$\langle c2 \rightarrow d3 \rightarrow f5 \rightarrow e7\rangle$	HIV	
5	$\langle d4 \rightarrow e6 \rightarrow f7 \rightarrow h8\rangle$	Diabetes	
6	$\langle d3 \rightarrow e5 \rightarrow h8\rangle$	Flu	
7	$\langle d3 \rightarrow e6 \rightarrow f7 \rightarrow h8\rangle$	Flu	

Table 9: Patient-specific path table P'

EPC	Path	Diagnosis	...
1	$\langle d(34) \rightarrow (ef)7 \rightarrow h8\rangle$	Diabetes	
2	$\langle d(34) \rightarrow (ef)5 \rightarrow h8\rangle$	HIV	
3	$\langle d(34) \rightarrow (ef)5 \rightarrow (ef)7 \rightarrow h8\rangle$	Flu	
4	$\langle d(34) \rightarrow (ef)5 \rightarrow (ef)7\rangle$	HIV	
5	$\langle d(34) \rightarrow (ef)6 \rightarrow (ef)7 \rightarrow h8\rangle$	Diabetes	
6	$\langle d(34) \rightarrow (ef)5 \rightarrow h8\rangle$	Flu	
7	$\langle d(34) \rightarrow (ef)6 \rightarrow (ef)7 \rightarrow h8\rangle$	Flu	

4.2.4 Extension to Global Generalization

In this subsection, we briefly discuss the privacy threats caused by various types of prior knowledge of attackers, and present an LKC-privacy preserved solution achieved by a sequence of generalizations and suppressions.

Example 4.2.1. A hospital would like to release the RFID card holders' moving path information, with their sensitive attributes in Table 8, to a third party data analyst. Each record contains a *path* and some patient-specific information, where a *path* contains a sequence of *pairs* $(loc_i t_i)$ indicating the patient's visited location loc_i at timestamp t_i. For example, $EPC\#3$ has a path $\langle d4 \rightarrow e5 \rightarrow f7 \rightarrow h8\rangle$, meaning that the patient has visited locations d, e f, and h at times 4, 5, 7, and 8, respectively. Without loss of generality, we assume that each data record contains only one sensitive value, namely HIV in Diagnosis. An attacker seeks to identify the record and/or sensitive value of a target victim from the published data.

Consider the following three scenarios:

Scenario I (Known location Attack). Suppose that the attacker knows that the target victim, Bob, has visited c and d. In Table 8, there are only three records, namely with EPC#2 and 4, that match locations c and d. Since these 2 records share the same sensitive diagnosis "HIV"; therefore, the attacker can infer that Bob has "HIV" with 100% confidence.

Scenario II (Known Time Attack). Suppose that the attacker knows that the target victim, Alice, has visited some unknown locations at time 2 and time 5. In Table 8, there is only one record, namely with EPC#5, that matches time 2 and 5. Therefore, the attacker can uniquely identify Alice's visited locations and her sensitive information.

Scenario III (Known location and Time Attack). Suppose that the attacker knows that the target victim, Kate, has visited location a at time 1. In Table 8, there is only one record, namely with EPC#2, that matches $a1$. Therefore, the attacker can uniquely identify Kate's visited locations and her sensitive information.

Figure 5 shows two user-defined taxonomy trees for generalizing locations and times. In this example, we use a full domain generalization scheme [19] [20] [18] [17] [29] [42]. In this scheme, if a child value v is generalized to its parent value, all instances of child values under the same parent value will be generalized to the parent value according to the user-defined taxonomy tree. For example, in Figure 5, if a is generalized to ab, then all instances of a and b are generalized to $\langle ab \rangle$, but c and d, which are the child values of $\langle cd \rangle$, remain unchanged.

Table 9 shows an example of anonymous table T' that satisfies $(2, 2, 50\%)$-privacy. The pairs $a1$ and $c2$ are suppressed, time 3 and 4 are generalized to $\langle 34 \rangle$ and location e and f are generalized to $\langle ef \rangle$. Every possible subsequence q with maximum length 2 is shared by at least 2 records and the confidence of inferring the sensitive value "HIV" from q is not greater than 50%. ■

Chapter 5

Experimental Results

The objective of the experiments presented in this chapter is to evaluate (1) the effectiveness of our proposed privacy model and anonymization method on preserving information utility, (2) the efficiency, and the scalability of the proposed anonymization algorithm. The effectiveness in information preservation is measured by comparing the difference of information utility before and after the anonymization. The efficiency is evaluated by the runtime of the test cases. The scalability is evaluated by runtime on some extremely large synthetic data. All experiments are conducted on a PC with Intel Core2 Quad 2.4GHz with 2GB of RAM. Unless otherwise specified, all experiments on our proposed method use Equation 1 as the $Score$ function.

5.1 RFID Data Generator

We developed a RFID Data Generator (RDG) to generate an object-specific data table for the purpose of the experiment. RDG considers the following data characteristics:

- RFID data must be placed in time sequence.
- The RDG must be flexible to change numbers of location and time in each record.

- The RDG can generate high-dimensional, sparse data sets.
- The RDG can generate different sensitive and insensitive diagnoses corresponding to each record.
- RFID data size is large enough to test the scalability of the proposed algorithm.

Based on the above characteristics, we develop a program that allows us to generate a high-dimensional and sparse data set that covers most of the RFID moving object patterns.

Algorithm 3 RFID Data Generator

Input: Length of the location L
Input: Length of the time T
Input: Size of the database S.
Output: Subway RFID raw database T.

```
1:  i = 0;
2:  j = 0;
3:  for all i < S do
4:      Pathlength = Randomlength;
5:      for all j < S do
6:          time = Randomtime;
7:          while time is within the timearray do
8:              time = Randomtime;
9:          end while
10:         Add time to timearray;
11:         ++j;
12:     end for
13:     ++i;
14: end for
15: Sort timearray;
16: k = 0;
17: for all k < Pathlength do
18:     Station = random Station;
19:     Time = timearray[k];
20:     Node = ConstructNode(Station,Time);
21:     Append Node to the Path;
22: end for
23: Diagnosis = random Diagnosis;
24: Career = random Career;
25: Record = ConstructRecord(Path, Diagnosis, Career);
26: Save Record to the database;
```

Algorithm 3 summarizes the RFID Data Generator. Lines 1-2 define parameters. Lines

3-14 generate a fixed size for the RFID path database. Each path has a randomly generated path length. For each path, the algorithm generates a random time. Line 15 sorts the time for each path in ascending order. At each iteration in Lines 17-22, the algorithm constructs a node that consists of a random subway station appended with a chronological time. At the end, the algorithm constructs a record that combines the path, diagnosis and career and saves the record to the database.

The employed data set is a simulation of the travel paths of 20,000 passengers over 24 hours in a subway system with 26 stations. Each record in the object-specific path table represents one passenger's path. There are 26×24 possible pairs, forming 576 dimensions. We simulate real-life travel patterns as follows. Most people travel to approximately 4 locations at different times during a day. For example, a student may leave home to go to school in the morning. After school, she/he goes to have lunch then goes home. A worker leaves home for work in the morning. He/she may go to the grocery store after work and then go home. Based on these types of patterns, we generate 16,000 passengers with a maximum path length of 4 pairs and 3,500 passengers with a maximum path length of 6 pairs. Some people, such as sales representatives and insurance agents, have to travel a lot. Thus, we consider 500 passengers who have a maximum path length of 26 pairs. Each record contains a disease attribute which has 5 possible values $\langle HIV, Flu, Headache, Diabetes, Handicap \rangle$. We considered one of them, namely HIV, to be the sensitive information in our experiments. The dataset characteristics are presented at Table 10.

5.2 Quality of Anonymous Data

Our first experiment is to measure the data quality of the LKC-privacy protected table T'. We use distortion ratio to measure the information loss caused by suppression. Let $N(T)$ and $N(T')$ be the total number of pair instances in tables T and T', respectively.

Table 10: Dataset Characteristics

	No. Path	Max Path 24	Max. Pairs	Number of Location	Number of Time	Diagnosis
T1	10,000	24	624	26	24	5
T2	20,000	24	624	26	24	5

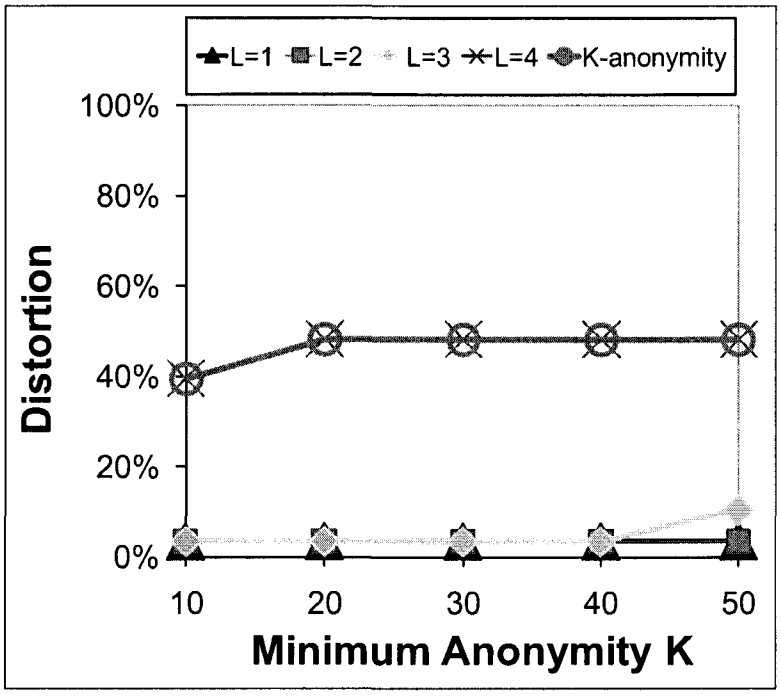


Figure 6: Distortion ratio vs. K where $C = 20\%$

The *distortion ratio*, computed by $\frac{N(T)-N(T')}{N(T)}$, measures the percentage of pair instances suppressed for achieving a given LKC-privacy requirement. Higher distortion ratio means lower data quality. We also compare our method with the traditional K-anonymization method.

Figure 6 depicts the distortion ratio of our method for maximum length $1 \leq L \leq 3$ for anonymity thresholds $10 \leq K \leq 50$ at confidence threshold $C = 20\%$, and compares the result with the traditional K-anonymity. In general, the distortion ratio is insensitive to the increase of K and stays between 3% to 10% for $1 \leq L \leq 3$ because this requirement only needs every sequence with a maximum length of 3 to be shared by at least 50 records out of 20,000 records. Comparing to traditional K-anonymity which consistently stays

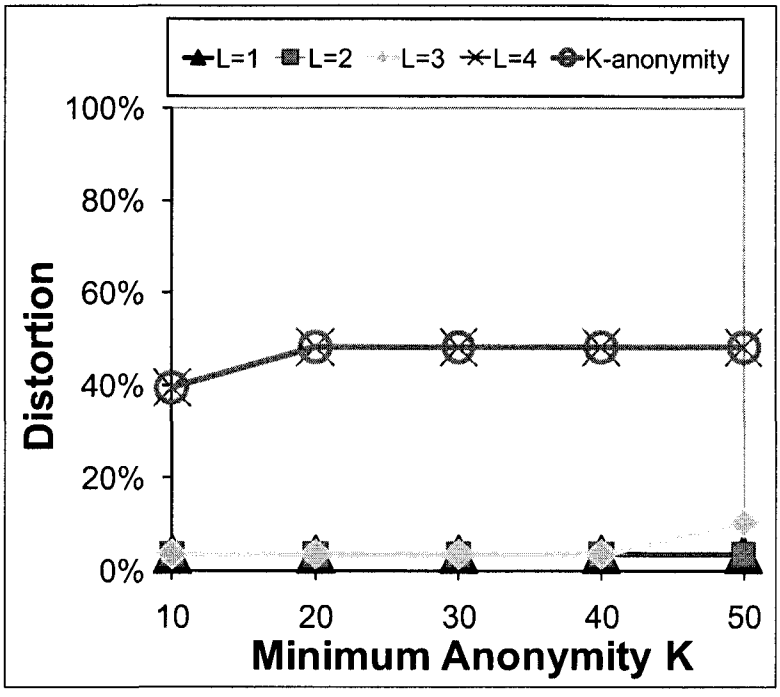


Figure 7: Distortion ratio vs. K where $C = 60\%$

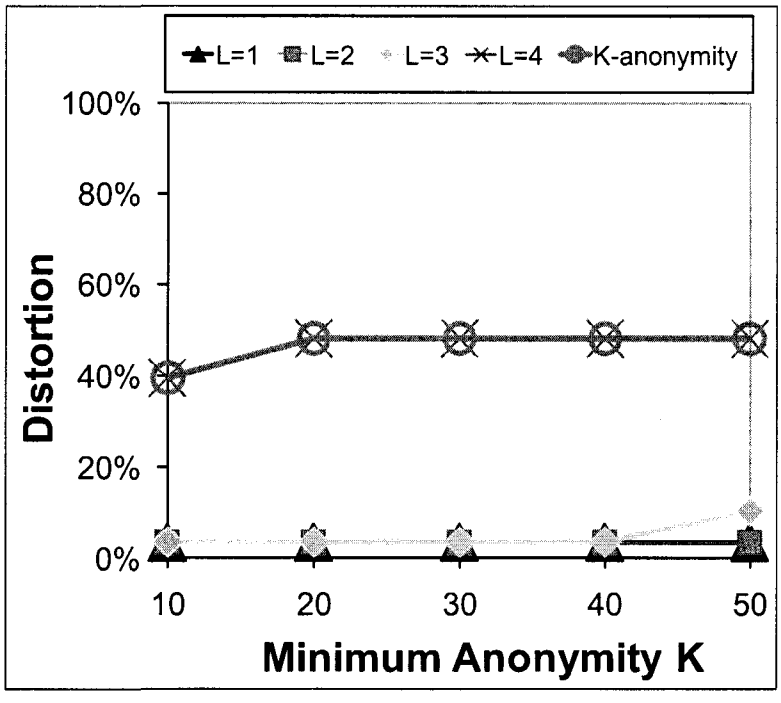


Figure 8: Distortion ratio vs. K where $C = 100\%$

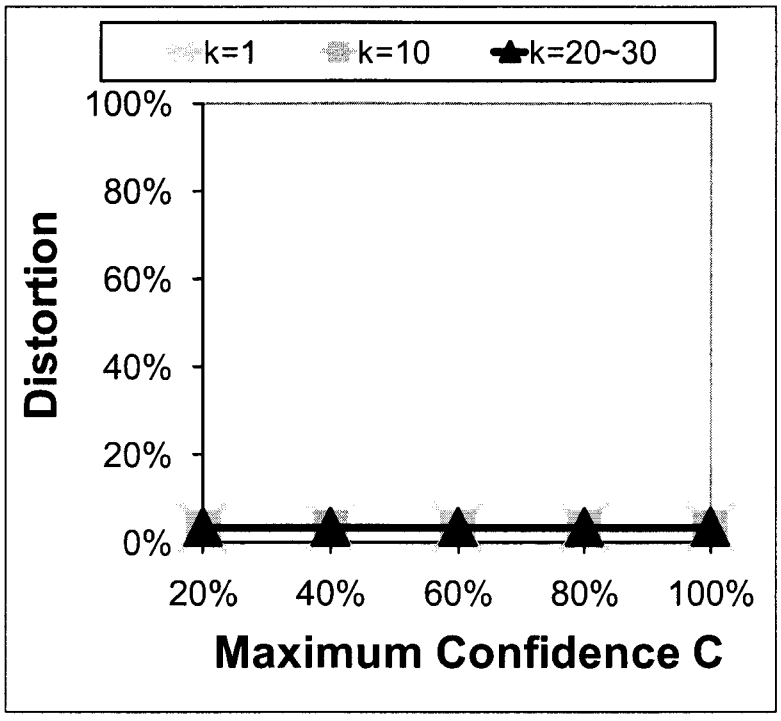


Figure 9: Distortion ratio vs. C where $L = 2$

above 40%, our anonymization method can effectively reduce information loss on high-dimensional data. As L increases to 4, the distortion ratio increases significantly because the majority of records have a path length of 4 pairs. Therefore, setting $L = 4$ yields a similar result to traditional K-anonymity. It is also interesting to note that the distortion ratio is insensitive to the change in confidence threshold C, implying that the primary driving force for suppressions is LK-anonymity, not LC-dilution. This fact is also reflected in Figure 7 and Figure 8 at $C = 60\%$ and $C = 100\%$, which is equivalent to ignoring LC-dilution because out of the 5 diagnoses $\langle HIV, Flu, Headache, Diabetes, Handicap \rangle$, only one diagnosis named HIV is considered sensitive and it is small percentage among the 20,000 records.

Figure 9 and Figure 10 compare the distortion ratio of our method with traditional K-anonymity for anonymity thresholds $10 \leq K \leq 30$ at confidence thresholds $20\% \leq C \leq 100\%$. At $L = 2$, in general, the distortion ratio is insensitive to the increase of

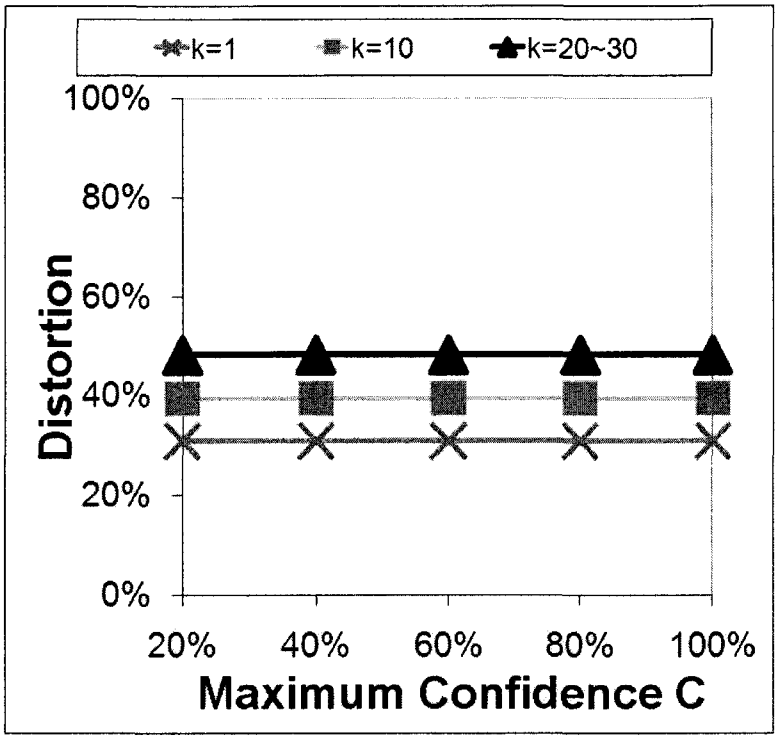


Figure 10: Distortion ratio vs. C where $L = infinity$

C and K and stays at around 3% because this requirement needs every sequence with a maximum length of 3 to be shared by at least 50 records out of 20,000 records. This figure illustrates two points. First, the distortion ratio is low, suggesting that the information is well-preserved even after achieving LKC-privacy. Second, setting $K = 1$ is equivalent to ignoring LK-anonymity and achieving only LC-dilution. This result once again confirms that LC-dilution has little affect on distortion ratio because the sensitive values are not particularly skewed in any particular group $G(q)$. In other words, it costs very little, or even nothing, to remove the privacy threats caused by attribute linking. Figure 11 compares the distortion ratio of the three $Score$ functions discussed in Chapter 4.2.1 for $10 \leq K \leq 50$ at $C = 60\%$. Experimental results suggest that the $Score$ function in Equation 1 yields the lowest distortion because it considers both privacy gain and information loss in its selection criteria. Figure 10 depicts the traditional K-anonymity when $L = infinity$. As seen, the distortion ratio is sensitive to the change in K but insensitive to C which consistently stays

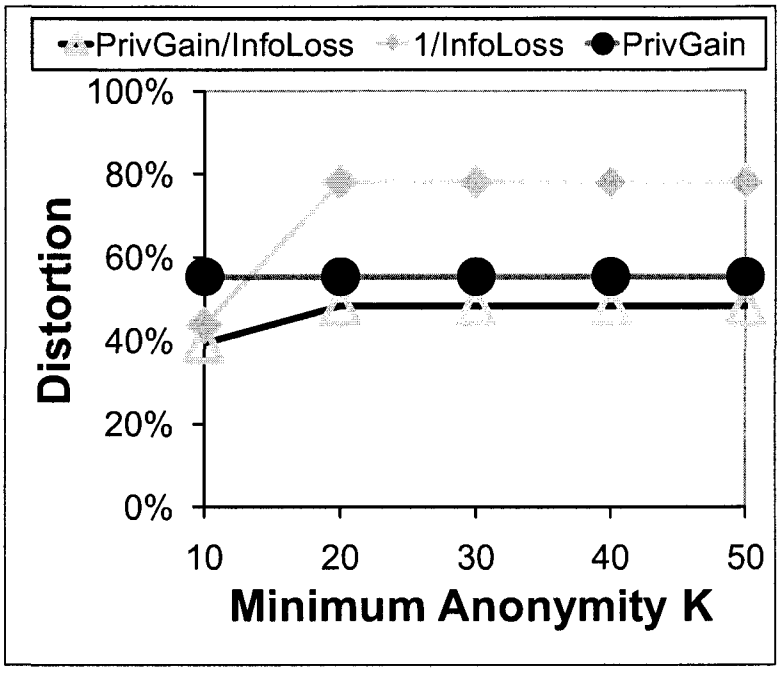


Figure 11: Distortion ratio vs. C with different $Score$

between 31% to 48%. As K increases to 30, the distortion rate reaches 48% and almost half of the data will be suppressed by the traditional K-anonymity method. It further proves that our LKC-model significantly reduces the distortion ratio while preserving the privacy compared to the traditional K-anonymity method.

Figure 12 shows the distortion ratio at length $1 \leq L \leq 4$. At $1 \leq L \leq 3$, the distortion is 3% but at $L = 4$, the distortion significantly increases to 42%. It further proves that our model significantly reduces the information loss for high-dimensional data. In addition, most of the records have a path length of 4 pairs. Therefore, $L = 4$ yields the similar result as traditional K-Anonymity.

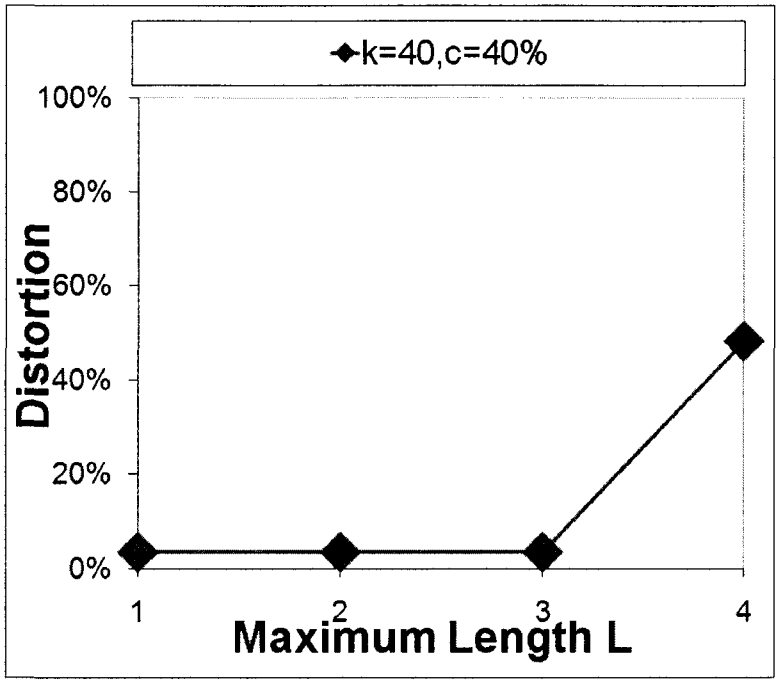


Figure 12: Distortion ratio vs. L

5.3 Efficiency and Scalability Analysis

Next, we examine the efficiency and scalability of our proposed anonymization method. For all the test cases conducted in Chapter 4, our method takes less than 1 second to complete. In an effort to further evaluate the scalability of our method, we conducted an experiment on some extremely large synthetic RFID data sets.

Figure 13 depicts the runtime in seconds from 200,000 records to 1 million records for $L = 3$, $K = 30$, $C = 60\%$. The total runtime for anonymizing 1 million records is 76 seconds, where 60 seconds are spent on identifying critical violations and 16 seconds are spent on reading the raw data file and writing the anonymous file. Thanks to the effective critical violation tree (CVT) data structure, the program takes less than 1 second to suppress all violations.

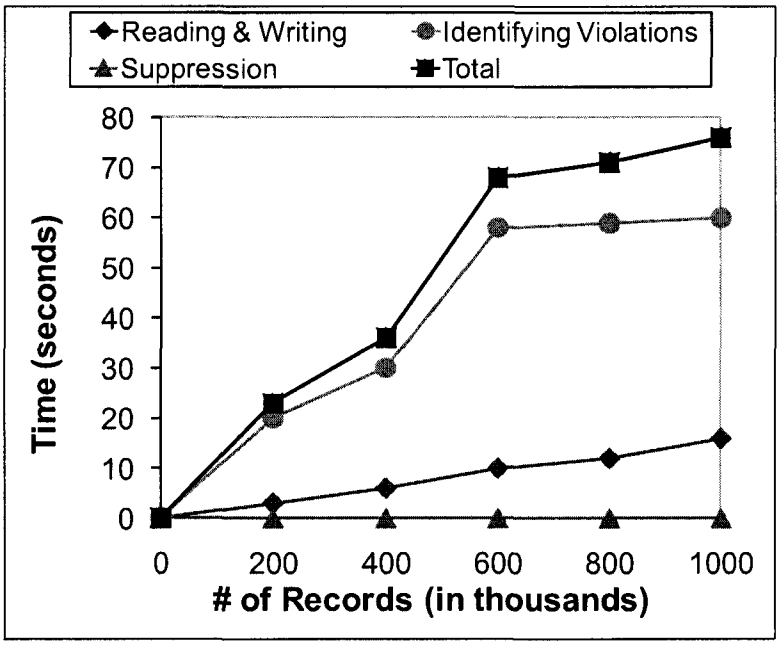


Figure 13: Scalability ($L = 3$, $K = 30$, $C = 60\%$)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

RFID is a promising technology applicable in many areas, but many of its privacy issues have not yet been addressed. In this thesis, we illustrate the privacy threats caused by publishing RFID data, formally define a privacy model, called LKC-privacy, for high-dimensional, sparse RFID data, and propose an efficient anonymization algorithm to transform a RFID data set to satisfy a given LKC-privacy requirement. We demonstrate that applying traditional K-anonymity on high-dimensional RFID data would render the data useless due to the curse of high-dimensionality. Moreover, we generate a RFID data set to model the subway system RFID database. Experimental results suggest that our method can efficiently anonymize large RFID data sets with significantly better data quality than the traditional K-anonymity method and significantly reduce the information loss compared to the traditional K-anonymity.

6.2 Future Work

The LKC-privacy model is a promising tool for preserving privacy and preventing information loss in high-dimensional, sparse data, especially for sequential data. In fact, our privacy model is a powerful one, and can be implemented in different applications. Our future work can be summarized as follows

- The LKC-privacy model is applicable to anonymize transaction data and sequential data, such as credit card data, cell phone data, Global Position System (GPS) data, and healthcare data [42].

- In this thesis, we use suppression techniques to achieve the best scalability of large databases. In the future, we can consider a combination of generalization and suppression techniques to preserve data relevancy.

- We applied a greedy algorithm in this thesis to achieve efficiency in the entire program. In the future, an optimal solution can be developed to achieve minimal information loss.

- In this thesis, we consider only the scenario of general publishing of RFID data. More specific data mining purposes, such as classification and clustering, can be considered to further reduce the information loss by applying $Score$ functions.

Bibliography

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 376–385, April 2008.

[2] N. R. Adam and J. C. Wortman. Security control methods for statistical databases. *ACM Computing Surveys*, 21(4):515–556, December 1989.

[3] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proc. of the 31st Very Large Data Bases (VLDB)*, pages 901–909, 2005.

[4] C. C. Aggarwal and P. S. Yu. On privacy-preservation of text and saprse binary data with sketches. In *Proc. of SIAM International Conference on Data Mining(SDM)*, Minneaoilis. MN, 2007.

[5] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of the 20th ACM PODS*, pages 247–255, Santa Barbara, CA, May 2001.

[6] R. Agrawal and R. Srikant. Privacy preserving data mining. In *Proc. of ACM SIGMOD*, pages 439–450, Dallas, Texas, May 2000.

[7] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, pages 217–228, Tokyo, Japan, 2005.

[8] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 1:46–55, 2003.

[9] R. Brand. Microdata protection through noise addition. In *Proc. of Control in Statistical Databases From Theory to Practice*, pages 97–116, London, United Kingdoms, 2002.

[10] Y. Bu, A. W. Fu, R. C. Wong, L. Chen, and J. Li. Privacy preserving serial data publishing by roled composition. In *Proc. of VLDB 2008*, Auckland, New Zealand, August 2008.

[11] J. Domingo-Ferrer and V. Torra. A critique of k-anonymity and some of its enhancements. In *Proc. of the 3rd International Conference on Availability, Reliability and Security (ARES)*, pages 990–993, 2008.

[12] W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proc. of 9th ACM SIGKDD*, Washington, DC, August 2003.

[13] E. Beinat. Privacy and location-based: Stating the policies clearly, 2001. GEO Informatics.

[14] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of 8th ACM SIGKDD*, pages 217–228, Edmonton, AB, Canada, July 2002.

[15] W. A. Fuller. Masking procedures for microdata disclosure limitation. *Official Statistics*, 9(2):383–406, 1993.

[16] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, In Press.

[17] B. C. M. Fung, K. Wang, L. Wang, and M. Debbabi. A framework for privacy-preserving cluster analysis. In *Proc. of the 2008 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Taipei, Taiwan, June 2008.

[18] B. C. M. Fung, K. Wang, L. Wang, and P. C. K. Hung. Privacy-preserving data publishing for cluster analysis. *Data & Knowledge Engineering (DKE)*, 68(6):552–575, June 2009.

[19] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, pages 205–216, Tokyo, Japan, April 2005.

[20] B. C. M. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(5):711–725, May 2007.

[21] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 2007.

[22] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 715–724, April 2008.

[23] H. Gonzalez, J. Han, and X. Li. Flowcube: Constructing RFID flowcubes for multi-dimensional analysis of commodity flows. In *Proc. of the International Conference on Very Large Data Bases (VLDB)*, pages 1–19, Seoul, Korea, September 2006.

[24] H. Gonzalez, J. Han, and X. Li. Mining compressed commodity workflows from massive RFID data sets. In *Proc. of the International Conference on Information and Knowledge Management (CIKM)*, November 2006.

[25] H. Gonzalez, J. Han, and X. Li. Mining compressed commodity workflows from massive RFID data sets. In *Proc. of CIKM 2006*, Arlington, Virginia, November 2006.

[26] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. pages 31–42, 2003.

[27] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Elsevier, 2 edition, 2006.

[28] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proc. of ACM SIGMOD*, pages 37–48, Baltimore, ML, 2005.

[29] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proc. of the 8th ACM SIGKDD*, pages 279–288, Edmonton, AB, Canada, July 2002.

[30] A. Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381– 394, February 2006.

[31] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proc. of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 99–106, Melbourne, FL, 2003.

[32] J. Kim and W. Winkler. Masking microdata files. In *Proc. of the ASA Section on Survey Research Methods*, pages 114–119, 1995.

[33] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of ACM SIGMOD*, pages 49–60, Baltimore, ML, 2005.

[34] J. Li, Y. Tao, and X. Xiao. Preservation of proximity privacy in publishing numerical sensitive data. In *Proc. of the ACM Conference on Management of Data(SIGMOD)*, pages 437–486, Vancouver, Canada, 2008.

[35] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity abd ℓ-diversity. In *Proc. of 21st IEEE International Conference on Data Engineering(ICDE)*, Istanbul, Turkey, 2007.

[36] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, April 2007.

[37] M. Barbaro and T. Zeller. A face is exposed for aol searcher no. 4417749, August 9 2006. New York Times.

[38] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proc. of the 22nd IEEE International Conference on Data Engineering (ICDE)*, 2006.

[39] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 1(1), March 2007.

[40] B. Malin and E. Airoldi. The effects of location access behavior on re-identification risk in a distributed enviroment. In *Proc. of 6th Workshop on Privacy Enhancing Technology(PET)*, pages 413–429, 2006.

[41] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of the 23rd ACM PODS*, pages 223–228, Paris, France, 2004.

[42] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. K. Lee. Anonymizing healthcare data: A case study on the red cross. In *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1285–1294, Paris, France, June 2009. ACM Press.

[43] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung. Privacy-preserving data mashup. In *Proc. of the 12th International Conference on Extending Database*

Technology (EDBT), pages 228–239, Saint-Petersburg, Russia, March 2009. ACM Press.

[44] S. Papadimitriou, S. Li, G. Kollios, and P. Yu. Time series compressibility and privacy. In *Proc. of the 33rd International Conference on Very Large Data Based (VLDB)*, pages 459–470, 2007.

[45] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proc. of the 17th ACM PODS*, page 188, June 1998.

[46] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. Technical Report SRI-CSL-98-04, SRI International, March 1998.

[47] S. E. Sarma, S. A. Weis, and D. W. Engels. RFID systems and security and privacy implications. In *Proc. of the 4th International Workshop of Cryptographic Hardware and Embedded Systems (CHES)*, pages 1–19, San Diego, 2003.

[48] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):571–588, 2002.

[49] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proc. of the 9th International Conference on Mobile Data Management (MDM)*, pages 65–72, April 2008.

[50] M. Terrovitis, N. Mamoulis, and P. Kalnis. Anonymity in unstructured data. Technical Report TR-2004-04, Department of Computer Science, University of Hong Kong, April 2008.

[51] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. In *Proc. of VLDB*, August 2008.

[52] The RFID Knowledgebase. Oyster transport for london tfl, card uk, January 2007. http://rfid.idtechex.com/knowledgebase/en/casestudy.asp?freefromsection=122.

[53] T. Trojer, B. C. M. Fung, and P. C. K. Hung. Service-oriented architecture for privacy-preserving data mashup. In *Proc. of the 7th IEEE International Conference on Web Services (ICWS)*, Los Angeles, CA, July 2009. IEEE Computer Society Press.

[54] K. Wang, B. C. M. Fung, and G. Dong. Integrating private databases for data analysis. In *Proc. of the 2005 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 171–182, Atlanta, GA, May 2005.

[55] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *Proc. of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 466–473, Houston, TX, November 2005.

[56] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowledge and Information Systems (KAIS)*, 11(3):345–368, April 2007.

[57] S.-W. Wang, W.-H. Chen, C.-S. Ong, L. Liu, and Y. Chuang. RFID applications in hospitals: a case study on a demonstration RFID project in a taiwan hospital. In *Proc. of the 39th Hawaii International Conference on System Sciences*, 2006.

[58] R. C. W. Wong, J. Li., A. W. C. Fu, and K. Wang. (α,k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proc. of the 12th ACM SIGKDD*, 2006.

[59] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. of ACM SIGMOD*, Chicago, IL, 2006.

[60] Y. Xu, B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Publishing sensitive transactions for itemset utility. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, November 2008.

[61] Y. Xu, K. Wang, A. W. C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In *Proc. of the 14th ACM SIGKDD*, August 2008.

[62] R. Yarovoy, F. Bonchi, L. Lakshmanan, and W. H. Wang. Anonymizing moving objects: How to hide a mob in a crowd? In *Proceedings of EDBT*, March 2009.

[63] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate query answering on anonymized tables. In *Proc. of the 23rd IEEE International Conference on Data Engineering (ICDE)*, April 2007.